International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 08 | Aug 2020 www.irjet.net p-ISSN: 2395-0072

Camera based Forward Collision Avoidance System

Mr. Prafull V. Barpute, Prof. B. H. Pansambal[2]

[1]ME Student, (E & TC) Signal processing, TSSM's BSCOER, Pune, Maharashtra, India.
[2] Professor, (E & TC) Signal processing, TSSM's BSCOER, Pune, Maharashtra, India

---

Abstract - *Car accidents and fatalities are a measure concern in today's world. Most of the accidents on highways are of type "front impact" which is because of un-ability of driver to identify / judge the front vehicle's movement due to lack of attention while driving. This project aims to address this issue by recognizing risk ahead of time and provide warning to user about the risk. Till today lot of work has already happened to tackle this problem and some of the projects are even implemented in high-end vehicles like VOLVO, BMW etc. Most of these projects use stereo vision algorithms and machine learning algorithms which are extensively resource hungry. So, these algorithms need expensive platforms like DSP or multi-core processors to implement and work at real time. Hence such products become expensive and out of reach from low range or mid range vehicle users. Current project will try to use simple Mono-vision algorithm to achieve same result without using resource hungry libraries. It does not depend on machine learning techniques hence not needed to have huge learning dataset. Project is written in C programming language and uses OpenCV as image processing library, CMake as building tool so as to be portable on any embedded platform. Project design includes design blocks like segmentation, blob extraction and validation and tracking on top of which FCW application works. This project provides valid audio-visual warning when any collision risk is identified in daylight conditions. As the project needs single camera and simple algorithm, it can be deployed on any low cost ARM controllers like Raspberry Pi. Hence overall product cost can be in within reach of low or mid range car users.*

Key Words: **FCW, Sensors, TTC, Tracking, Segmentation**

1. INTRODUCTION

Passive safety system is the safety system that helps to protect occupant of car in case of accident happens. So, passive safety systems are reactive systems. In contrary, active safety systems are proactive in nature. Purposes of such systems are to avoid accident at all. Active safety systems are always running which includes monitoring and analysis of the environment and taking decision appropriately. Active safety systems implemented on embedded platform should also use optimum power consumption so as to become battery friendly. Some of the active systems give only visible/audible warning to drivers while others responsible to control the driving also. There are many active safety systems in use currently like RCTA, BSD, LCA, FCTA etc. Forward Collision Warning (FCW) system is also an active safety system.

FCW system senses the environment i.e. portion of road in front of driver, using sensors. Sensors can be visual sensors like monocular / stereo cameras or other sensors like RADAR and LiDAR. This project uses monocular camera as sensor for FCW system. Limitation of any FCW system depends on sensor which is used. Some FCW systems use sensor fusion technique to overcome limitations of individual sensors which increases system complexity as well. Normally vision sensors are costlier than RADAR because of processing power demand but fails in low light conditions. This approach tries to address this shortcoming by implementing simple low computationally complex algorithms.

2. Literature Survey

There have been enormous efforts to develop a variety of FCW algorithms. The previous researches reviewed in this study are classified into three categories, including time based method, stopping distance based method and Artificial Neural Network (ANN) based method.

2.1 Time based methods

One of the most widely used in the time based methods is Time-To-Collision (TTC), which is designed to consider a time that would be taken for the collision risk between preceding and following vehicle.

$$TCC = \frac{\Delta d}{\Delta v} \quad (1)$$

Δd is the distance between host vehicle and immediate proceeding Δv vehicle and is relative velocity between two vehicles. The TTC is measured under the assumption that the preceding vehicle will maintain current acceleration before it stops. This is a perpetual based warning algorithm witch trigger warnings based on the time required for two vehicles to collide if they continue traveling at their current speed and path. When the "Time to Collision" (TTC) falls below the human perceptual threshold a warning is activated to alert the driver. [1]

2.2 Size based Methods

Width of a vehicle in image is proportional to real width of the vehicle according to the pinhole camera geometry. If real width of a vehicle is known, range to the vehicle can be calculated as in the following:

$$F_c d = \frac{W_a}{w_a} \quad (2)$$

Where F_c is focal length of camera and w_a and W_a are vehicle width in image and vehicle real width, respectively. Vehicle real width varies from 1.4m to 2.6m. Applying this formula for range estimation without prior knowledge of vehicle real width may introduce significant error, which can be as much as 30%, if a fixed width (e.g., W_a = 1.82 m) is used. It is not accurate enough for computing TTC, but it can be used as sanity check.[1]

2.3 Stopping distance based Methods

The main idea of stopping distance based methods to measure rear-end collision risk is that the stopping distance of the preceding vehicle should be greater than that of following vehicle for car-following situation [1]. Many algorithms have been proposed based on the concept [1]. This is also called as Kinematic based warning algorithm. Consistent with perceptual-based algorithms that rely exclusively on range and speed data, kinematic-based algorithms also use deceleration rate and reaction time data to determine the minimum theoretical distance to stop safely. However, unlike perceptual algorithms that depend on empirical knowledge of human perception, kinematic-based algorithms determines the distance at which braking onset must occur if a collision is to be avoided. [1]

The representative of Stopping Distance Algorithm (SDA) can be described

$$SDA = V_p t + \frac{V_p^2}{2a_p^{Dec}} + L - V_F \tau - \frac{V_F^2}{2a_F^{Dec}} \quad (3)$$

t is the time headway between two consecutive vehicles,

a_P^{Dec} is deceleration rate of the preceding vehicle,

a_F^{Dec} is deceleration rate of the following vehicle,

τ is following vehicle's Perception-Reaction Time (PRT).

It is safe when the SDA value is greater than 0, while it is not safe when the SDA value is less than 0. In the SDA, all of variables are observable in the V2I communication environment, except for PRT. The PRT is an important parameter and vital component in the success of the rear-end CWS since it is a key factor to determine whether the current driving status is safe or not [5, 16]. The previous studies have been to address the braking behavior related problems performed based on the deterministic or stochastic parametric approaches. Most of the algorithms use the PRT value ranging from 0.5 to 2.0 (s) [1, 2]. Although the conservative PRT values seem to be more reasonable in terms of enhancing the safety performance, the deterministic PRT values may result in showing low performance level of such algorithms since there are some variations for individual PRT under the various traffic conditions. Therefore, the previous algorithms are not likely to be appropriate for applying to actual driving mode

2.3 Artificial Neural Network (ANN) based methods

Since the previous parametric deterministic approaches show the limitations that do not reflect the influence of PRT, many researchers have attempted to develop collision-warning strategies based on the non-parametric methods.

One of the most dominant emerging technologies on the rear-end CWS is Artificial Neural Network (ANN) based method. ANN based method has a merit in that it can deal with solving the unobservable complex problem. A vehicle control strategy for multi directional collision avoidance has initially been proposed [1, 3]. This study adopted Multi-Layer Perception Neural Network (MLPNN) and fuzzy logic algorithm respectively. More recently, an approach to combine the neural networks and fuzzy logic was intended to develop an algorithm that controls automobile-on-the highway based on the high accurate GPS data [4].

There are also some other FCW methods with stereo vision as well as with using other sensors like RADAR. However I am not considering them here because they are out of scope of this project.

All above described methods (and there various versions) have one or more limitations. Time based method is most widely used but it has serious errors as it does not consider curvature of road. Hence False Positive and False Negative rate is high. This is more adverse in case of night time.

SDA method is dependent on PRT value which is not constant under all driving conditions as varies even from driver to driver. Although the conservative PRT values seem to be more reasonable in terms of enhancing the safety performance, the deterministic PRT values may result in showing low performance level of such algorithms since there are some variations for individual PRT under the various traffic conditions.

ANN based method is require large amount of initial training data and performance is unknown in case of unknown / not trained scenarios.

Proposed method is extension of time based method where curvature of road is also taken in consideration.

3. System Description in details

This section describes in detail the design approach used in current project.

3.1 Terminologies Used

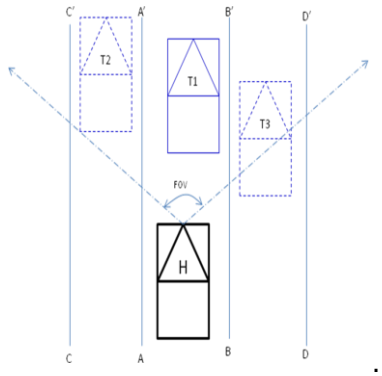

Fig -1: Terminology Used

H indicates host vehicle. Algorithm is tested with various cameras with FOV within 36 to 55 degrees. T1 indicates target vehicle in the Ego lane.

T2 and T3 represents target vehicles in adjacent lanes AA' BB' are internal lane markings CC' and DD' are external lane markings

Objects outside Lane D and C are not considered for any metric calculations and completely ignored while generating warning.

3.2 Block Details

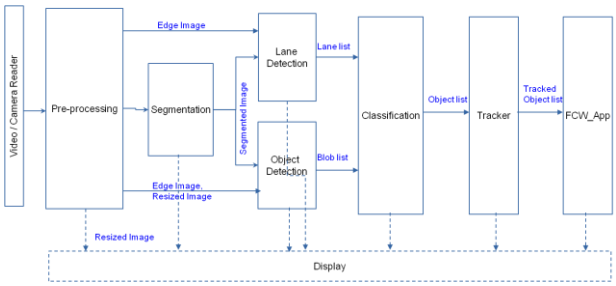

Fig -2: Block Diagram of FCW System

Video / Camera Reader : This module is responsible for reading frames / images from HDD or capturing frames from camera feed. This module is controllable by using macro definition.

Pre-processing : This module removes noise by Gaussian smoothing, convert color image to grayscale and then find edge map.

Segmentation : This module does segmentation of image. First it convert color image (BGR) to HSV model. Then extract ROI (Region Of Interest). Then apply threshold on the extracted ROI to get segmented image. It also finds out valid blobs from segmented image.

Lane Detection : It extracts the lines based on Hough Transform. It then filters extracted lines based on slope and some other properties. Then it calculates road / lane boundaries which is used further in classification module

Object Detection : In this module, openCV, simple blob detector algorithm is used to extract blobs, which are further used along with blobs extracted in segmentation stage.

Classification : In this module blob lists are filtered by various parameters like ROI filter, Lane filter etc. Similar or nearby blobs are merged and objects are formed by assigning confidence weight to them.

Tracker : In this module, objects are compared with their previous history and assigned lifetime value. Based on lifetime and confidence values final list of tracked objects are calculated. In this module distance and velocity parameters are calculated based on current and historical parameters.

FCW Application : This module responsible for generating Forward collision warning based on distance, speed, direction and position of each tracked object.

Display : This module is temporary module and used mainly for debugging purpose. It provides intermediate stages as well as final output on the display.

3.3 FCW Application Details

In order to determine the distance from camera to a known object. We are going to utilize triangle similarity[4]. The triangle similarity goes something like this: Let's say we have a marker or object with a known width W. We then place this marker some distance D from our camera. We take a picture of our object using our camera and then measure the apparent width in pixels P. This allows us to derive the perceived focal length F of our camera:

$$F = \frac{P \times D}{W} \quad (4)$$

So once we know the value of F for any particular camera, D for any object say D' can be calculated as below

$$D' = \frac{W \times F}{P} \quad (5)$$

Then relative velocity can be calculated by formula

$$V_{Rel} = \frac{D_{Prv} - D_{cur}}{T_f} \quad (6)$$

Time to collision is calculated from relative velocity and lateral distance.

$$TTC = \frac{D_L}{V_{Rel}} \quad (7)$$

Below is simple flowchart of FCW application. Based on TTC value calculated, warning can be of type "ALERT" means driver has to take some action or it can be of type "Informative" which driver can observe and take decision to take action or not. Else the "SAFE to DRIVE" option is continue to display.

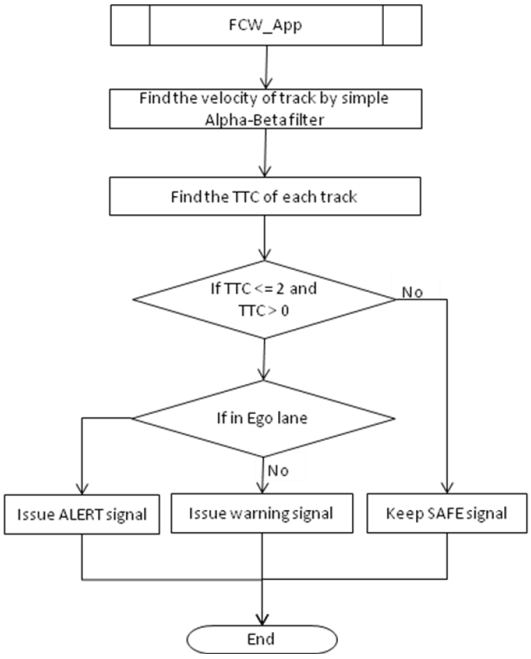

Fig -3: Flow chart of FCW application

4. TESTING

This project uses standard CMake tool to compile the project files and libraries.

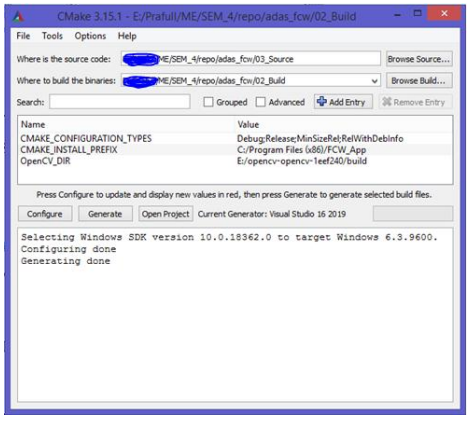

Fig -2: Test Environment

As the coding is done in C programming language, application is ported on Raspberry PI 4 board as embedded test platform. Offline testing is done on both Windows and Linux OS on laptop (Configuration: 64-bit 4GB RAM, Pentium ® 3558U @1.70 GHz)

4.1 Test Results

Fig -4: Intermediate stage Results
(a. Original Image b. Segmentation c. Labeling d. Object Detection f. Tracking)

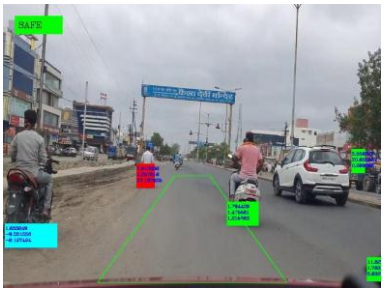

Fig -5: No threat of collision

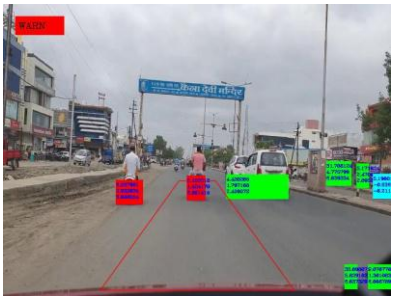

Fig -6: Collision warning

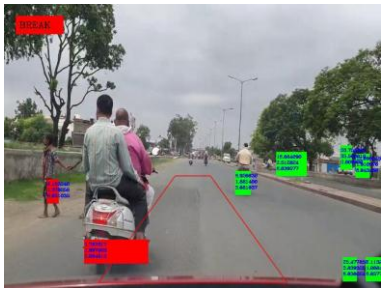

Fig -7: Collision Alert

Fig -8: Highway Scenario

4.2 Performance Testing Results

Table -1: Performance in terms of time (Frames per Second)

Sr. No.	Road Type		Traffic Density		Weather		Average FPS	
	Highway	Urban	High	Low	Sunny	Cloudy	Laptop	Embedded Board
1	X		X		X		23	21
2	X			X	X		20	21
3		X		X		X	10	12
4	X		X		X		21	22
5	X			X	X		25	25

Table -2: Performance in terms of accuracy

Sr. No.	Road Type		Traffic Density		Weather		False Positive	
	Highway	Urban	High	Low	Sunny	Cloudy	Laptop	Embedded Board
1	X		X		X		2	2
2	X			X	X		1	1
3		X		X		X	5	5
4	X		X		X		2	2
5	X			X	X		2	2

Table -3: Performance in terms of accuracy

Sr. No.	Road Type		Traffic Density		Weather		False Negative	
	Highway	Urban	High	Low	Sunny	Cloudy	Laptop	Embedded Board
1	X		X		X		1	1
2	X			X	X		0	0
3		X		X		X	3	3
4	X		X		X		0	0
5	X			X	X		0	0

5. Rationale

5.1 Features

- This is very lightweight algorithm and can work on real time.
- This algorithm does not require any training to be provided.
- This algorithm is written in plan C hence can be ported on any embedded platform. Currently it is tested on Raspberry PI 4 with Raspbian OS along with Linux and Windows.

5.2 Limitations

- Presently system works only in clear daylight conditions. Raining / Snowy conditions are not tested. Night detection is out of scope.
- In case of sensor is noisy, performance gets affected because of lane detection system. Trying to make noise removal more robust.
- In case of absence of lane marking, more false positives can appear.
- Obstacles other than vehicles (like bicycles, Pedestrian etc.) may not get accurate warning.

6. Conclusion

By analyzing output of the project, it can be concluded that algorithm developed is lightweight and can be easily ported on any low cost embedded platform. Most of similar systems present in market required costly DSP processors to do the same thing. Developed system is capable of providing valid warning to avoid accidents of type front collision.

REFERENCES

[1] https://www.researchgate.net/publication/279446898_A_Study_on_the_Rear-End_Collision_Warning_System_by_Considering_Different_Perception-Reaction_Time_Using_Multi-Layer_Perceptron_Neural_Network

[2] https://www.researchgate.net/publication/260371976_Robust_Range_Estimation_with_a_Monocular_Camera_for_Vision-Based_Forward_Collision_Warning_System

[3] https://www.mobileye.com/wp-content/uploads/2011/09/ForwardCollisionWarning

[4] https://www.pyimagesearch.com/2015/01/19/find-distance-camera-objectmarker-using-python-opencv/